UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/730,190	12/05/2000	Kestutis Patiejunas	MS160309.1	7993

27195 7590 08/19/2005

AMIN & TUROCY, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
ALI, SYED J	
ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 08/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/730,190	**Applicant(s)** PATIEJUNAS, KESTUTIS	
	Examiner Syed J. Ali	**Art Unit** 2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 May 2005*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-50* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. In view of the appeal brief filed on May 17, 2005, PROSECUTION IS HEREBY REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or,

(2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or other evidence are permitted. See 37 CFR 1.193(b)(2).

2. Claims 1-50 are presented for examination.

3. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. **Claims 1-34 and 46-50 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**

6. As per claims 1, 8, and 46, the claimed "software component" is non-statutory as it is not tangibly embodied, as the "software component" is implemented entirely in software. Claims 2-7, 9-22, and 47-50 are rejected for at least the same reasons as presented for their parent claims, as they fail to present any limitations that resolve the deficiencies of the claims from which they depend.

7. As per claim 23, the claim language raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a concrete, useful, and tangible result to form the basis of statutory subject matter under 35 U.S.C. 101. The claimed "method" should be modified to indicate that it is embodied in a manner as to be executable, e.g. "a computerized method" or "a computer-implemented method". Claims 24-34 are rejected for at least the same reasons as their parent claim, as they fail to present any limitations that resolve the deficiencies of the claim from which they depend.

Claim Rejections - 35 USC § 103

8. **Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert et al. (USPN 6,687,729) (hereinafter Sievert).**

9. As per claim 1, Sievert teaches the invention as claimed, including a client side HTTP stack software component for processing requests, comprising:

at least one completion port object (col. 3 lines 20-32);

a thread pool comprising a plurality of threads adapted to process tasks associated with at least one client side request (col. 3 lines 20-32); and

a client side state machine associated with the at least one request (col. 3 lines 34-65).

10. Sievert does not specifically limit the disclosure to a software component existing on the "client side." Rather, Sievert discusses a thread pool in general, where any computer that utilizes a thread pool or implements multi-threading may make use of the method for thread pool management, whether it is a client, server, or some other computer. A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art.

Additionally, the state machine disclosed by Sievert refers to the operation of the work queue, which in turn function with respect to individual threads, i.e. the work queue is the data structure by which individual threads are serviced and perform work. There is no limitation in the claims, either explicit or implicit, that prohibits intervening data structures to aid with the processing of requests.

11. **Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Jones et al. (USPN 6,003,061) (hereinafter Jones).**

12. As per claim 2, Jones teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

13. It would have been obvious to one of ordinary skill in the art to combine Sievert and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

14. **Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Okano et al. (USPN 6,725,253) (hereinafter Okano).**

15. As per claim 3, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a DNS thread adapted to resolve domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

16. It would have been obvious to one of ordinary skill in the art to combine Sievert and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

17. **Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia et al. (USPN 6,493,749) (hereinafter Paxhia).**

18. As per claim 4, Paxhia teaches the invention as claimed, including the client side HTTP stack implementation of claim 1, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list (col. 41 lines 19-28).

19. It would have been obvious to one of ordinary skill in the art to combine Sievert and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

20. **Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia as applied to claim 4 above, and further in view of Jones.**

21. As per claim 5, Jones teaches the invention as claimed, including the client side HTTP stack implementation of claim 4, farther comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

22. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

23. **Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia in view of Jones as applied to claim 5 above, and further in view of Okano.**

24. As per claim 6, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 5, further comprising a DNS thread adapted to resolve domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

25. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, Jones, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

26. **Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sievert in view of Paxhia as applied to claim 4 above, and further in view of Okano.**

27. As per claim 7, Okano teaches the invention as claimed, including the client side HTTP stack implementation of claim 4, further comprising a DNS thread adapted to resolve domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

28. It would have been obvious to one of ordinary skill in the art to combine Sievert, Paxhia, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

29. **Claims 8, 23, 35, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM Technical Disclosure Bulletin ("Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls") (hereinafter IBM).**

30. As per claim 8, IBM teaches the invention as claimed, including a software component for implementing a client side HTTP stack, comprising:

a thread pool comprising N threads adapted to process M requests from a client application component, wherein N and M are integers greater than 1 and wherein M is greater than N (pg. 199).

31. IBM discusses managing a thread pool for requests made by an application server, without explicitly indicating the thread pool is for use on the "client side". While the remote server may typically handle RPC calls, there is no reason to believe that the thread pool could not be implemented on the client side. A recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to

patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim. In this particular case, the thread pool of IBM could be easily implemented on the client side.

32. As per claim 23, IBM teaches the invention as claimed, including a method of implementing a client side HTTP stack, comprising:

processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N (pg. 199).

33. As per claim 35, IBM teaches the invention as claimed, including a computer-readable medium having computer-executable instructions for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N (pg. 199).

34. As per claim 46, IBM teaches the invention as claimed, including a software component for implementing a client side HTTP stack, comprising:

means for processing M requests from a client application component using a thread pool comprising N threads, wherein M and N are integers greater than 1 and wherein M is greater than N (pg. 199).

35. **Claims 9-13, 17-19, 24-28, 32-34, 36-39, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert.**

36. As per claim 9, Sievert teaches the invention as claimed, including the software component of claim 8, further comprising at least one thread activation component adapted to activate at least one of the N threads based on an event (col. 3 lines 45-52).

37. It would have been obvious to one of ordinary skill in the art to combine IBM and Sievert since the method of IBM is absent guidance as to how threads are handled in terms of sending and receiving data. IBM is limited to showing a method for initializing and controlling the size of a thread pool. Sievert provides additional functionality for a pool of threads to handle work requests as well as encapsulating requests and responses within an I/O completion port, thereby easing the manner in which requests are handled. The use of a completion port is beneficial in that it simplifies distributed computing for multiple concurrent requests by handling all incoming and outgoing data.

38. As per claim 10, Sievert teaches the invention as claimed, including the software component of claim 9, wherein the at least one thread activation component is a completion port (col. 3 lines 20-32).

39. As per claim 11, Sievert teaches the invention as claimed, including the software component of claim 9, wherein at least one of the N threads is adapted to deactivate itself and

return to the thread pool when an operation being processed by the at least one of the threads is pending (col. 5 lines 26-38).

40. As per claim 12, Sievert teaches the invention as claimed, including the software component of claim 11, wherein the event is the receipt of a completion packet by the at least one thread activation component (col. 3 lines 27-32).

41. As per claim 13, Sievert teaches the invention as claimed, including the software component of claim 12, wherein the at least one thread activation component is a completion port (col. 3 lines 20-32).

42. As per claim 17, Sievert teaches the invention as claimed, including the software component of claim 9, further comprising a state machine associated with at least one of the M requests (col. 3 lines 34-65).

43. As per claim 18, Sievert teaches the invention as claimed, including the software component of claim 17, further comprising at least one key associated with the at least one of the M requests, wherein a first one of the N threads is associated with the at least one of the M requests, and wherein the thread activation component is adapted to associate the context of the first one of the N threads with the at least one state machine using the at least one key, in order to activate the first one of the N threads (col. 5 line 59 - col. 6 line 54).

44. As per claim 19, Sievert teaches the invention as claimed, including the software component of claim 18, wherein the thread activation component is adapted to associate the context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

45. As per claim 24, Sievert teaches the invention as claimed, including the method of claim 23, further comprising:

selectively deactivating at least one of the N threads (col. 5 lines 26-38); and

activating at least another of the N threads based on an event using at least one thread activation component (col. 3 lines 45-52).

46. As per claim 25, Sievert teaches the invention as claimed, including the method of claim 24, wherein the at least one thread activation component is a completion port (col. 3 lines 20-32).

47. As per claim 26, Sievert teaches the invention as claimed, including the method of claim 24, wherein selectively deactivating at least one of the N threads comprises deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending (col. 5 lines 26-38).

48. As per claim 27, Sievert teaches the invention as claimed, including the method of claim 26, wherein activating at least another of the N threads based on an event comprises:

receiving a completion packet using the thread activation component (col. 3 lines 27-32); and

activating one of the N threads upon receipt of the completion packet using the thread activation component (col. 3 lines 45-52).

49. As per claim 28, Sievert teaches the invention as claimed, including the method of claim 27, wherein the at least one thread activation component is a completion port (col. 3 lines 20-32).

50. As per claim 32, Sievert teaches the invention as claimed, including the method of claim 26, further comprising associating a state machine with at least one of the M requests (col. 3 lines 34-65).

51. As per claim 33, Sievert teaches the invention as claimed, including the method of claim 32, further comprising:

associating at least one key with the at least one of the M requests (col. 5 line 59 - col. 6 line 54);

associating a first one of the N threads with the at least one of the M requests (col. 5 line 59 - col. 6 line 54); and

associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads (col. 5 lines 26-38; col. 5 line 59 - col. 6 line 54).

52. As per claim 34, Sievert teaches the invention as claimed, including the method of claim 33, further comprising associating a context of one of the N threads with the at least one state machine using the at least one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

53. As per claim 36, Sievert teaches the invention as claimed, including the computer-readable medium of claim 35, further comprising computer-executable instructions for:

selectively deactivating at least one of the N threads (col. 5 lines 26-38); and

activating at least another of the N threads based on an event using at least one thread activation component (col. 3 lines 45-52).

54. As per claim 37, Sievert teaches the invention as claimed, including the computer-readable medium of claim 36, wherein the at least one thread activation component is a completion port (col. 3 lines 20-32).

55. As per claim 38, Sievert teaches the invention as claimed, including the computer-readable medium of claim 36, wherein the computer-executable instructions for selectively deactivating at least one of the N threads comprises computer-executable instructions for deactivating the at least one of the N threads when an operation being processed by the at least one of the N threads is pending (col. 5 lines 26-38).

56. As per claim 39, Sievert teaches the invention as claimed, including the computer-readable medium of claim 38, wherein the computer-executable instructions for activating at least another of the N threads based on an event comprises computer-executable instructions for:

receiving a completion packet using the thread activation component (col. 3 lines 27-32); and

activating one of the N threads upon receipt of the completion packet using the thread activation component (col. 3 lines 45-52).

57. As per claim 47, Sievert teaches the invention as claimed, including the software component of claim 46, further comprising:

means for selectively deactivating at least one of the N threads (col. 5 lines 26-38); and

means for activating at least another of the N threads based on an event (col. 3 lines 45-52).

58. **Claims 14, 29, 40, and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claims 13, 28, 39, and 47 above respectively, and further in view of Jones.**

59. As per claim 14, Jones teaches the invention as claimed, including the software component of claim 13, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

60. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

61. As per claim 29, Jones teaches the invention as claimed, including the method of claim 28, further comprising activating an object scheduled to begin sending requests at a specific time using a scheduler thread (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

62. As per claim 40, Jones teaches the invention as claimed, including the computer-readable medium of claim 39, further comprising computer-executable instructions for activating an object scheduled to begin sending requests at a specific time using a scheduler thread (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

63. As per claim 48, Jones teaches the invention as claimed, including the software component of claim 47, further comprising means for activating an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

64. **Claims 15, 30, 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert in view of Jones as applied to claims 14, 29, and 40 above respectively, and further in view of Okano.**

65. As per claim 15, Okano teaches the invention as claimed, including the software component of claim 14, further comprising a DNS thread adapted to resolve domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

66. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, Jones, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

67. As per claim 30, Okano teaches the invention as claimed, including the method of claim 29, further comprising resolving domain names into IP addresses using a DNS thread (col. 12 line 37 - col. 13 line 5).

68. As per claim 41, Okano teaches the invention as claimed, including the computer-readable medium of claim 40, further comprising computer-executable instructions for resolving domain names into IP addresses using a DNS thread (col. 12 line 37 - col. 13 line 5).

69. **Claims 16, 31, and 42-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert in view of Jones in view of Okano as applied to claims 15, 30, and 41 above respectively, and further in view of Paxhia.**

70. As per claim 16, Paxhia teaches the invention as claimed, including the software component of claim 15, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list (col. 41 lines 19-28).

71. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, Jones, Okano, and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

72. As per claim 31, Paxhia teaches the invention as claimed, including the method of claim 30, further comprising selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket (col. 41 lines 19-28).

73. As per claim 42, Paxhia teaches the invention as claimed, including the computer-readable medium of claim 41, further comprising computer-executable instructions for

selectively timing out at least one socket according to at least one timer associated with the at least one socket using a timeout thread comprising a list of active sockets and timers associated with each socket (col. 41 lines 19-28).

74. As per claim 43, Sievert teaches the invention as claimed, including the computer-readable medium of claim 42, further comprising computer-executable instructions for associating a state machine with at least one of the M requests (col. 3 lines 34-65).

75. As per claim 44, Sievert teaches the invention as claimed, including the computer-readable medium of claim 43, further comprising computer-executable instructions for:

associating at least one key with the at least one of the M requests (col. 5 line 59 - col. 6 line 54);

associating a first one of the N threads with the at least one of the M requests (col. 5 line 59 - col. 6 line 54); and

associating a context of the first one of the N threads with the at least one state machine using the at least one key, in order to deactivate the first one of the N threads (col. 5 line 59 - col. 6 line 54).

76. As per claim 45, Sievert teaches the invention as claimed, including the computer-readable medium of claim 44, further comprising computer-executable instructions for associating a context of one of the N threads with the at least one state machine using the at least

one key in order to activate the one of the N threads based on an event (col. 5 line 59 - col. 6 line 54).

77. **Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Jones.**

78. As per claim 20, Jones teaches the invention as claimed, including the software component of claim 8, further comprising a scheduler thread adapted to activate an object scheduled to begin sending requests at a specific time (col. 19 lines 39-49; col. 20 line 62 - col. 21 line 6).

79. It would have been obvious to one of ordinary skill in the art to combine IBM and Jones since the prescheduling of threads allows the resource usage of a system to be known at compile time rather than run time. Particular advantages can be achieved in terms of load balancing and resource utilization by providing particular information related to the start time of an operation in advance. Additionally, the setting of a particular start time is beneficial to real time systems that have threads with hard deadlines or other scheduling constraints.

80. **Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Okano.**

81. As per claim 21, Okano teaches the invention as claimed, including the software component of claim 8, further comprising a DNS thread adapted to resolve domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

82. It would have been obvious to one of ordinary skill in the art to combine IBM and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

83. **Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Paxhia.**

84. As per claim 22, Paxhia teaches the invention as claimed, including the software component of claim 8, further comprising a timeout thread with a list of active sockets and timers associated with each socket, and adapted to selectively timeout at least one socket according to at least one timer in the list (col. 41 lines 19-28).

85. It would have been obvious to one of ordinary skill in the art to combine IBM and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

86. **Claim 49 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claim 47 above, and further in view of Okano.**

87. As per claim 49, Okano teaches the invention as claimed, including the software component of claim 47, further comprising means for resolving domain names into IP addresses (col. 12 line 37 - col. 13 line 5).

88. It would have been obvious to one of ordinary skill in the art to combine IBM, Sievert, and Okano since IP addresses are expressed in octets that make it difficult to remember domain names. Rather, easy to remember domain names are provided that are then translated into IP addresses easing the use of a networked system by a user (Okano, col. 2 lines 4-10).

89. **Claim 50 is rejected under 35 U.S.C. 103(a) as being unpatentable over IBM in view of Sievert as applied to claim 47 above, and further in view of Paxhia.**

90. As per claim 50, Paxhia teaches the invention as claimed, including the software component of claim 47, further comprising means for selectively timing out at least one socket according to at least one timer associated with the at least one socket (col. 41 lines 19-28).

91. It would have been obvious to one of ordinary skill in the art to combine IBM and Paxhia since a thread that has been operating for an extended period of time without responding may be causing a starvation condition. The use of a timer to monitor a socket ensures that a thread does not stall while utilizing one of the system's sockets. The expiration of the timer thus alarms the

system that the thread should be terminated, thereby protecting system resources and ensuring that other threads receive a fair share of the processor.

Response to Arguments

92. **Applicant's arguments with respect to claims 1-50 have been considered but are moot in view of the new grounds of rejection.**

Conclusion

93. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
August 16, 2005

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100